

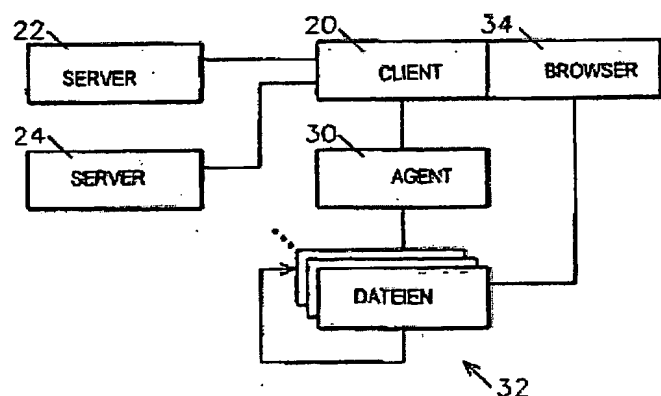
Access provision method for web server, involves grouping web pages which contain static link addresses in accordance with unchanged web pages

Patent number: DE10051021
Publication date: 2001-06-28
Inventor: BENNETT CRAIG A (US); BERANEK MICHAEL J (US); BONASSO JEFFREY J (US); DAVIA CHRISTOPHER J (US); STAPEL STEVEN W (US)
Applicant: IBM (US)
Classification:
- **International:** G06F17/30; G06F3/00
- **European:** G06F17/30W9
Application number: DE20001051021 20001014
Priority number(s): US19990427459 19991022

Report a data error here

Abstract of DE10051021

Amendment in web page with dynamic connection is performed based on determined dynamic linkage address for every possible combination of dynamic connection on web page. Web pages containing static link addresses are grouped in accordance with unchanged web pages. Independent claims are also included for the following: (a) Method for access content of web page; (b) Data processing method; (c) Server HTML language converting method; (d) An agent; (e) System for resolution and storage of dynamic connection; (f) Computer program product





①⑨ BUNDESREPUBLIK
DEUTSCHLAND



DEUTSCHES
PATENT- UND
MARKENAMT

①⑫ **Offenlegungsschrift**
①⑩ **DE 100 51 021 A 1**

⑤① Int. Cl. 7:
G 06 F 17/30
G 06 F 3/00

②① Aktenzeichen: 100 51 021.3
②② Anmeldetag: 14. 10. 2000
④③ Offenlegungstag: 28. 6. 2001

③⑩ Unionspriorität:
09/427459 22. 10. 1999 US
⑦① Anmelder:
International Business Machines Corp., Armonk,
N.Y., US
⑦④ Vertreter:
Kauffmann, W., Dipl.Phys. Dr., Pat.-Ass., 70569
Stuttgart

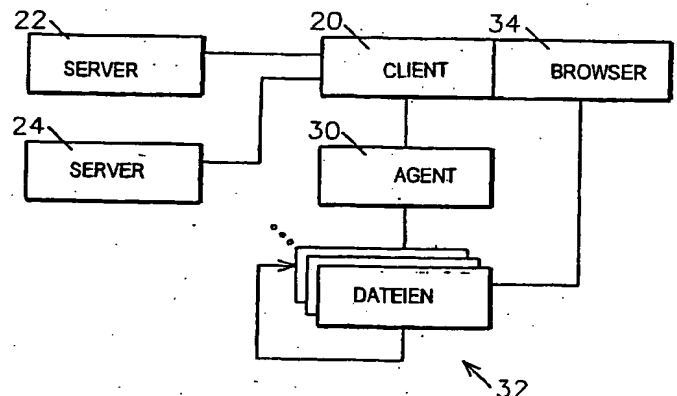
⑦② Erfinder:
Bennett, Craig A., Woodstock, Ga., US; Beranek,
Michael J., Marietta, Ga., US; Bonasso, Jeffrey J.,
Pittsburgh, Pa., US; Davia, Christopher J., Smyrna,
Ga., US; Stapel, Steven W., Eden Prairie, Minn., US

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

Der Inhalt dieser Schrift weicht von den am Anmeldetag eingereichten Unterlagen ab
Prüfungsantrag gem. § 44 PatG ist gestellt

⑤④ System, Verfahren und Computerprogramm zur Veröffentlichung interaktiver Web-Inhalte in einer statisch verknüpften Web-Hierarchie

⑤⑦ Während ein Client mit einem Server verbunden ist, simuliert ein Agent-Tool die interaktive Benutzeraktion und die Durchforstung dynamischer Web-Seiten, wodurch die Prozesse auf Server-Seite HTML-Seiten an den Client senden. Die auf diese Weise bereitgestellten Seiten werden vom Agent aufgenommen. Anschließend fügt der Agent statische Hyperlinks ein, mit denen die gerichtete Navigationslogik auf Server-Seite ersetzt wird. Schließlich werden die auf diese Weise bereitgestellten Seiten permanent in lokalen Dateien gespeichert, wo sie für die Offline-Navigation durch einen Browser zur Verfügung stehen, ohne dass auf den oder die Browser zugegriffen werden muß. Die HTML-Seiten werden dabei weitergeleitet.



DE 100 51 021 A 1

DE 100 51 021 A 1

Beschreibung

Hintergrund der Erfindung

Fachgebiet der vorliegenden Erfindung

Die vorliegende Erfindung bezieht sich im allgemeinen auf die Web-Technology und im besonderen auf die Auflösung und Speicherung dynamischer Verknüpfungen als statische Verknüpfungen zur Veröffentlichung von Web-Inhalten.

Hintergrund

Web-Browser, die sich offline befinden, laden Web-Seiten mit allen dazugehörigen Grafiken usw. herunter und speichern sie auf lokale Medien, beispielsweise auf die Festplatte eines Client, wo man sie offline betrachten kann. Viele dieser Browser ermöglichen es einem Benutzer, eine "Tiefe" (maximale Anzahl einzuhaltender rekursiver Verknüpfungen) anzugeben, und stellen auf den vom Benutzer angegebenen Seiten Verknüpfungen zu anderen Seiten her. Das ist für Benutzer hilfreich, denen online nur eine begrenzte Zeit zur Verfügung steht oder die eine besonders gute Web-Seite mit Grafiken usw. speichern möchten.

Auf diesem Gebiet gibt es zahlreiche Client-Web-Cache-Programme, beispielsweise GetBot, Robo Surfer, Web Buddy, WebCopy99, WebZip und Surf Express. Darüber hinaus sind viele Web-Server auf der Server-Seite mit einer Cache-Funktion ausgestattet, die in etwa so funktioniert wie die Web-Cache-Programme auf der Client-Seite.

Keines dieser Cache-Programme bietet jedoch die Möglichkeit, datengesteuerte dynamische Verknüpfungen herzustellen, Verknüpfungen, die abgeleitet werden, indem auf dem Server eine Logik ausgeführt wird, möglicherweise in Verbindung mit Parametern, die in einer Interaktion zwischen Benutzer und einer Web-Seite übertragen werden, die ursprüngliche Seite so abzuändern, dass alle benötigten Daten enthalten sind, darunter auch statische Verknüpfungen und Javascript, um zu Seiten zu gelangen, zu denen Verknüpfungen hergestellt wurden.

Ein Ziel der vorliegenden Erfindung besteht darin, in einem Prozeß auf der Client-Seite ein System und ein Verfahren zur Veröffentlichung dynamisch verknüpfter interaktiver Web-Inhalte in einer statisch verknüpften Web-Hierarchie bereitzustellen.

Ein weiteres Ziel der vorliegenden Erfindung besteht darin, ein System und ein Verfahren zur Erkennung der Struktur einer Web-Site bereitzustellen und alle dynamisch erzeugten Inhalte in statische Seiten umzuwandeln.

Ein weiteres Ziel der vorliegenden Erfindung besteht darin, ein System und ein Verfahren bereitzustellen, um Funktionskomponenten in dynamisch verknüpften interaktiven Web-Seiten so zu ändern, dass sie sich auf der Client-Seite gleich verhalten, ohne dass auf der Server-Seite Transaktionen verarbeitet werden müssen.

Ein weiteres Ziel der vorliegenden Erfindung besteht darin, ein System und ein Verfahren zur Veröffentlichung hoch interaktiver Web-Inhalte auf einem verteilbaren Medium bereitzustellen, wodurch eine Server- oder Netzwerkverbindung überflüssig wird.

Ein weiteres Ziel der vorliegenden Erfindung besteht darin, ein System und ein Verfahren für den interaktiven Zugriff auf den Web-Inhalt im Offline-Modus oder dort, wo die Netzwerk-Infrastruktur eine Verbreitung auf lokalen Medien erfordert, bereitzustellen.

Ein weiteres Ziel der vorliegenden Erfindung besteht darin, ein System und ein Verfahren zur Veröffentlichung der Inhalte von HTML-Seiten bereitzustellen, die auf der Grundlage einer interaktiven Aktion durch den Benutzer von einem Web-Server dynamisch erstellt wurden, als ob diese Daten interaktiv geladen wurden. Ein weiteres Ziel der vorliegenden Erfindung besteht darin, die daraus resultierenden Inhalte über lokale Medien verfügbar zu machen.

Ein weiteres Ziel der vorliegenden Erfindung besteht darin, ein System und ein Verfahren für den Zugriff auf die Inhalte von HTML-Seiten bereitzustellen, die auf der Grundlage einer interaktiven Aktion durch den Benutzer von einem Web-Server dynamisch erstellt wurden, ohne dass eine Verbindung zum Server aufgebaut ist.

Ein weiteres Ziel der vorliegenden Erfindung besteht darin, ein System und ein Verfahren zur Veröffentlichung der Inhalte von Web-Seiten auf einer CD-ROM oder auf einem anderen Speichermedium beim Client sowie für den Zugriff auf diese Inhalte über einen nicht angeschlossenen Computer-Browser bereitzustellen.

Ein weiteres Ziel der vorliegenden Erfindung besteht darin, ein System und ein Verfahren bereitzustellen, das dynamische Verknüpfungen ermöglicht, die auf der Server-Seite auf der Programmlogik Java oder Common Gateway Architecture (CGA) beruhen.

Ein weiteres Ziel der vorliegenden Erfindung besteht darin, ein System und ein Verfahren bereitzustellen, das in der Lage ist, datengesteuerte dynamische Verknüpfungen herzustellen und die ursprüngliche Seite so abzuändern, dass sie alle für den Zugriff auf die verknüpften Seiten erforderlichen Daten enthält.

Ein weiteres Ziel der vorliegenden Erfindung besteht darin, ein System und ein Verfahren bereitzustellen, das es einem Client-Web-Cache-Programm ermöglicht, datengesteuerte dynamische Verknüpfungen herzustellen, Verknüpfungen, die abgeleitet werden, indem auf dem Server eine Logik ausgeführt wird, möglicherweise in Verbindung mit Parametern, die in einer Interaktion zwischen Benutzer und einer Web-Seite übertragen werden, die ursprüngliche Seite so abzuändern, dass alle benötigten Daten enthalten sind, darunter auch statische Verknüpfungen und Javascript, um zu Seiten zu gelangen, zu denen Verknüpfungen hergestellt wurden.

Ein weiteres Ziel der vorliegenden Erfindung besteht darin, ein System und ein Verfahren für die Umwandlung einer HTML-Gruppe (Hyper-Text Markup Language) bereitzustellen, für die eine interaktive Server-Verbindung zu einer HTML-Gruppe erforderlich ist, die keine Server-Interaktion erfordert.

Zusammenfassung der vorliegenden Erfindung

In Übereinstimmung mit dem Verfahren und dem System der vorliegenden Erfindung wird eine HTML-Web-Seite

(Hyper-Text Markup Language) von einem Agent auf dynamische Verknüpfungen durchsucht, für die es erforderlich ist, dass der Server eine nächste HTML-Gruppe erzeugt. Diese dynamischen Verknüpfungen werden daraufhin durch errechnete statische Darstellungen in einer oder mehreren Dateien auf einem Festspeicher ersetzt, wo sie für einen Browser zur Verfügung stehen.

In Übereinstimmung mit einem ersten Aspekt der vorliegenden Erfindung wird ein Computerprogramm bereitgestellt, das so konfiguriert ist, dass es dynamische HTML-Verknüpfungen durch errechnete statische Darstellungen ersetzen kann.

Weitere Merkmale und Vorteile der vorliegenden Erfindung werden aus der nachfolgenden ausführlichen Beschreibung des derzeit bevorzugten Ausführungsbeispiels der vorliegenden Erfindung ersichtlich, die durch die beigefügten Zeichnungen ergänzt wird.

Kurze Beschreibung der Zeichnungen

Fig. 1 ist ein übersichtliches Systemdiagramm eines Server/Client-Systems, das einen Agent gemäß dem bevorzugten Ausführungsbeispiel der vorliegenden Erfindung umfaßt.

Fig. 2 ist eine Diagrammdarstellung einer Web-Seite, die eine 1-zu-1-Abbildung von dynamischer zu statischer Verknüpfung zeigt.

Fig. 3 ist eine Diagrammdarstellung einer auf einem Server erzeugten Web-Seite, auf der Auswahlkombinationen ersichtlich sind, sowie einer mit einem entsprechenden Agent erstellten Web-Seite, die das Aussehen und die Benutzerführung der ursprünglichen auf dem Server erzeugten Web-Seite beibehält.

Fig. 4 ist eine Diagrammdarstellung einer mit einem Agent erzeugten Web-Seite, die auf der vom Server erstellten Web-Seite von Fig. 3 beruht, wobei das Aussehen und die Benutzerführung abgeändert sind.

Fig. 5 ist eine Darstellung mehrerer hierarchisch verknüpfter Web-Seiten.

Fig. 6 ist ein Flußdiagramm des Verfahrens gemäß der vorliegenden Erfindung zur Beibehaltung des Aussehens und der Benutzerführung der auf dem Server erzeugten Web-Seite.

Fig. 7 ist ein Flußdiagramm des mit dem Agent ausgeführten Verfahrens der vorliegenden Erfindung, das es ermöglicht, das Aussehen und die Benutzerführung der auf dem Server erzeugten Web-Seite entweder zu ändern oder beizubehalten.

Beste Anwendung für die vorliegende Erfindung

Wir betrachten zunächst Fig. 1. Hier wird das System eines bevorzugten Ausführungsbeispiels der vorliegenden Erfindung dargestellt. Der Client 20 ist mindestens mit einem Server 22 oder aber mit mehreren Servern, darunter Server 24, verbunden. Das Verfahren der vorliegenden Erfindung ist im Agent 30 implementiert, der sich, wie aus der Darstellung hervorgeht, im Client 20 befindet. Dieser Agent kann jedoch auch Server-gestützt sein (nicht dargestellt). Der Agent 30 erzeugt in mehreren Datendateien und/oder Logikdateien 32 statische Darstellungen dynamischer Server-gestützter Verknüpfungen. Wenn der Client 20 von den Servern 22, 24 getrennt ist, greift der Browser 34 auf die Dateien 32 zu, um eine interaktive Aktion des Benutzers und den Zugriff auf dynamische Web-Seiten anhand dieser statischen Darstellungen der dynamischen Verknüpfungen zu simulieren.

In Übereinstimmung mit dem bevorzugten Ausführungsbeispiel der vorliegenden Erfindung kann, wenn der Client 20 an den Server 22 angeschlossen ist, das Agent-Tool 30 eine interaktive Aktion des Benutzers und den Zugriff auf dynamische Web-Seiten simulieren, was dazu führt, dass die Prozesse 22, 24 auf der Server-Seite HTML-Seiten an den Client 20 liefern können. Sobald diese Seiten bereitgestellt werden, werden sie vom Agent 30 erfasst, durch statische Hyperlinks ergänzt, wodurch die gerichtete Navigationslogik auf der Server-Seite ersetzt wird, und in den Dateien 32 permanent gespeichert, wo sie für die Offline-Navigation durch den Browser 34 zur Verfügung stehen, so dass die Server 22, 24 überflüssig sind. Die HTML-Seiten werden auf diese Weise veröffentlicht.

Wir betrachten als nächstes die Fig. 2 und 3. Hier werden von den vielen in Frage kommenden Arten von Web-Seiten zwei Arten veranschaulicht.

Zunächst betrachten wir Fig. 2, die Diagrammdarstellung einer Web-Seite, die eine 1-zu-1-Abbildung von dynamischer zu statischer Verknüpfung zeigt. Diese Web-Seite fordert einen Benutzer auf, unter den Möglichkeiten A, B und C eine Antwort auszuwählen. Wie aus diesem Beispiel hervorgeht, ist die Antwort "ONLY A" die einzig richtige. Folglich wird die Antwort "ONLY A" durch die Verknüpfung 79 mit einer Seite 78 verknüpft, wo dem Benutzer eine korrekte Antwort angezeigt wird. Alle anderen Antworten werden durch die Verknüpfung 77 mit einer Seite 76 verknüpft, wo dem Benutzer die falschen Antworten angezeigt werden. Diese Verknüpfungen 77, 79 sind dynamische Verknüpfungen, die am Server 22 aufgelöst werden. Der Agent 30 fragt die Auflösung des Servers ab und hartkodiert die entsprechenden Verknüpfungen 77, 79, die er empfängt, als statische Verknüpfungen im HTML, das in den Dateien 32 gespeichert ist.

Die abgeleiteten Verknüpfungen 77, 79 auf der Server-Seite werden durch entsprechende hartkodierte Verknüpfungen 77, 79 ersetzt, während die vom Server erzeugten Antwortseiten 76, 78 in die lokalen Dateien 32 kopiert werden. Der Server 22 weiß beispielsweise, dass, wenn der Benutzer die Option "ONLY A" auswählt, eine dynamische Verknüpfung zur richtigen Antwortseite 78 hergestellt und dem Benutzer angezeigt wird. Ansonsten wird eine dynamische Verknüpfung zu Seite 76 mit den falschen Antworten hergestellt. Der Agent 30 HTML sendet eine Nachricht an den Server 22 und fordert darin die Auflösung der Verknüpfung an. Diese Auflösung der Verknüpfung wird in den Dateien 32 vom Agent 30 durch eine Verknüpfung zur Seite mit der richtigen Antwort in den Dateien 32 ersetzt.

Wir betrachten nun Fig. 3. Dargestellt wird eine Web-Seite, auf der eine Entscheidung getroffen werden muß. Die richtige Antwort wird durch eine Kombination von Auswahlmöglichkeiten dargestellt. Jede beliebige Kombination aus Null, A, B, C und D ist eine mögliche Antwort, und eine oder mehrere dieser beliebigen Kombinationen kann die richtige Antwort sein. Die richtigen Antworten nehmen im Quadrat zur Anzahl (N) der möglichen Auswahlen zu (wenn Null keine gültige Antwort darstellt, ist dieser Wert $2^N - 1$) und sind nur A, nur B, nur C, nur D, A und B, A und C, A und D, B und

C, B und D, C und D usw. (Alle möglichen 2^N Antworten einschließlich Null werden in Fig. 4 dargestellt.) Fig. 3 stellt sowohl eine Server-gestützte Web-Seite (die ohne Referenz zum Agent der Erfindung erzeugt wurde) als auch eine Client-gestützte Web-Seite, die aus der Verarbeitung des Agent 30 hervorgeht, dar. Dabei wird das Aussehen und die Benutzerführung der ursprünglichen auf dem Server erzeugten Web-Seite beibehalten. Aussehen und Benutzerführung beibehalten bezieht sich in diesem Fall auf die Beibehaltung der ABCD-Antworten gemäß Darstellung in Fig. 3 und nicht auf das Ersetzen der vom Server erzeugten Anzeige von Fig. 3 durch eine Anzeige, die alle möglichen Kombinationen aufführt, die in eine statische Form umgewandelt werden könnten, wie dies in Fig. 4 geschieht.

In einem ersten Beispiel erfolgt die Erzeugung einer Clientgestützten Web-Seite gemäß Darstellung in Fig. 4 ohne Beibehaltung des Aussehens und der Benutzerführung durch Öffnen einer URL-Verbindung zum Server 22. Als Antwort wird eine Gruppe von HTML-Seiten (Web-Seiten) empfangen. Für weitere Verweise an diesen Server 22 werden diese HTML-Seiten gescannt. Für jeden Verweis an einen Server wird eine URL-Verbindung geöffnet. Danach werden weitere Antworten in Form weiterer HTML-Seiten empfangen. Dies wird für jeden weiteren Verweis wiederholt. Bei jeder Antwort, die empfangen wird, werden in eine flache Datei 32 entsprechende Web-Seiten mit statischen Verknüpfungen geschrieben, die die Struktur von URL-Verweisen in der ursprünglichen HTML-Seite beibehalten.

In einem zweiten Beispiel erfolgt die Erzeugung einer Clientgestützten Web-Seite gemäß Darstellung in Fig. 3 ohne Änderung des Aussehens und der Benutzerführung (d. h. ohne dass die Anzahl der angezeigten möglichen Auswahlen geändert wird) ungefähr so wie im ersten Beispiel, außer dass, da Ergebnisse vom Server 22 empfangen werden, jeder potenzielle Eingang simuliert wird, um alle möglichen Ergebnisse abzuleiten. Diese Ergebnisse können in einer Wahrheitstabelle 73 gesammelt, in einem Array oder als verknüpfte Liste, flache Datei oder Hash-Tabelle dargestellt oder implementiert werden, wie dies beispielsweise in Javascript der Fall oder einem anderen Objekt von Client 20 beispielsweise im Hauptspeicher (nicht dargestellt) der Fall ist. Diese werden wiederum in den lokalen Dateien 32 mit den Antwortseiten 75 verknüpft. (Java wird unter der Adresse <http://www.javasoft.com> beschrieben; Javascript wird unter der Adresse <http://devedge.netscape.com> beschrieben, und CGI wird unter der Adresse <http://www.cgi-perl.com> beschrieben.)

Wenn in Beispiel 1 eine Wahrheitstabelle verwendet wird, um Antworten mit Antwortseiten zu verknüpfen, ist es nicht erforderlich, dass die Verknüpfungen in den HTML-Seiten noch vorhanden sind. Stattdessen werden, wie aus der Darstellung in Fig. 4 hervorgeht, Änderungen neue Anzeigerauswahloptionen zugeordnet, die alle möglichen Änderungen repräsentieren. Ein Beispiel für eine solche Wahrheitstabelle ist in Tabelle 1 enthalten.

TABELLE 1: WAHRHEITSTABELLE

	A	B	C	D
A	0	0	0	0
B	0	0	1	0
C	0	1	0	0
D	0	0	0	0

Die Wahrheitstabelle im obigen Beispiel gibt an, dass die Antworten von C und B richtig und alle anderen falsch sind. Das Javascript auf der Client-Seite prüft die Antwort des Benutzers, vergleicht diese Antwort mit den in der Wahrheitstabelle enthaltenen gültigen Werten und legt den zu folgenden Pfad fest. Die Wahrheitstabelle wird vom Agent 30 während der Verarbeitung erstellt und in die resultierende Web-Datei 32 auf der Client-Seite eingefügt.

Wir betrachten als nächstes Fig. 4. Das Aussehen und die Benutzerführung der Web-Seite wird dahingehend geändert, dass alle möglichen Antworten aufgelistet werden. Der Agent 30 kann aber auch die Logik auf der Server-Seite 22 mit einer Logik auf der Server-Seite 20 (beispielsweise einem Applet) ersetzen, um die Vorgänge auf der Server-Seite zu emulieren. Ein solches Applet behandelt bestimmte Fragentypen und eine ganz bestimmte Logik auf der Server-Seite, beispielsweise Multiple Choice und eine Antwort, Multiple Choice und mehrere Antworten, Vergleiche, Tabellen usw. Der Agent 30 liefert Parameter an den Server, die in den erzeugten Web-Inhalt eingebettet sind und den Server veranlassen, die Logik auf der Client-Seite bereitzustellen.

Wir betrachten nun Fig. 5. Hier wird eine Hierarchie von Web-Seiten dargestellt. In diesem Beispiel wird die Seite 50 vom Server 22 bedient und enthält Verknüpfungen mit den Seiten 52 und 54, die auch vom Server 22 bedient werden. Seite 54, die beispielsweise mit Seite 70 übereinstimmt, enthält Verknüpfungen mit den Seiten 62, 64 und 66, wobei beispielsweise die Seite 64 mit der Antwortseite 76 und die Seite 66 mit der Antwortseite 78 übereinstimmt. Seite 62 enthält eine Verknüpfung mit Seite 68, die in diesem Beispiel vom Server 24 bedient wird. Die Seiten 52 und 54 haben eine Tiefe von 1, die Seiten 62, 64 und 66 haben eine Tiefe von 2, und die Seite 68 hat eine Tiefe von 3 unter der ursprünglichen oder Elternseite 50. Da die Server 22 und 24 diese Seiten an den Client 20 leiten, wenn die Aufforderung dazu vom Agent 30 kommt, wird jede Seite in eine separate Datei 32 gestellt und werden die dynamischen Verknüpfungen mit den Servern 22, 24 durch statische Verknüpfungen zwischen den entsprechenden Seiten in den Dateien 32 ersetzt.

In einem anderen bevorzugten Ausführungsbeispiel kann die Seite 54 mit der Seite 72 und die Seiten 64 und 66 mit den Antwortseiten 75 übereinstimmen. Die resultierende Wahrheitstabelle 73 befindet sich an einer Stelle im Hauptspei-

cher, die vorübergehend zur Herstellung von Verknüpfungen zwischen der Seite 72 und den Antwortseiten 75 verwendet wird.

Wir betrachten nun Fig. 6. Hier wird ein erstes bevorzugtes Ausführungsbeispiel des Verfahrens der vorliegenden Erfindung, das vom Agent 30 ausgeführt wird, beschrieben. Angenommen, die Benutzereingabe erfolgt an www.ibm.com/hr/index.html, wobei der Server 22 repräsentiert wird durch www.ibm.com. Die IP-Adresse ist 9.243.100.100.

In Schritt 80 empfängt der Client 20 die IP-Adresse 9.243.100.100 vom Server www.ibm.com 22.

In Schritt 82 wird die nächste URL-Variable auf die gesamte Adresse www.ibm.com/hr/index.html eingestellt.

In Schritt 84 wird eine URL-Verbindung zum nächsten URL geöffnet.

In Schritt 86 fordert der Agent 30 den Inhalt dieser URL-Verbindung an und speichert ihn ab.

In Schritt 88 durchsucht der Agent 30 die Inhalte des Speichers und gruppiert alle Verweise auf den Server 22 oder auf die IP-Adressen oder andere Adressen anderer Server 24.

In Schritt 90 werden die Schritte 84–88 an allen Verweisen wiederholt, wobei der Agent 30 seine Aufforderungen an die Server 22, 24 sendet und den daraufhin empfangenen Inhalt in den neuen lokalen Dateien 32 speichert und in den Schritten 84 bis 88 für jeden Durchlauf die nächste URL aktualisiert.

Im Schritt 92 verarbeitet der Agent 30 die gespeicherten Dateien 32 und ersetzt die Server-Verweise (Verknüpfungen mit den Servern 22, 24) durch lokale Dateiverweise (Verknüpfungen mit anderen Dateien aus der Gruppe von Dateien 32).

Das Ergebnis ist eine Gruppe lokaler Dateien 32, die einander und nicht die Server 22, 24 referenzieren.

Wir betrachten als nächstes Fig. 7. Gezeigt wird eine Alternativimplementierung des vom Agent 30 ausgeführten Verfahrens, das es ermöglicht, das Aussehen und die Benutzerführung der an den Browser übergebenen Web-Seiten entweder zu ändern oder beizubehalten.

In Schritt 100 erhält der Agent 30 die IP-Adresse des Servers 22.

In Schritt 102 wird das nächste URL mit der Quelldatei www.ibm.com/hr/index.html gleichgesetzt.

In Schritt 104 wird eine URL-Verbindung zum nächsten URL geöffnet.

In Schritt 106 fordert der Agent 30 den Inhalt dieses URL an und speichert ihn ab.

In Schritt 108 durchsucht der Agent 30 die Inhalte des Speichers und gruppiert (kennzeichnet) alle Verweise auf den Server 22 (also www.ibm.com/hr/index.html oder die IP-Adresse 9.243.100.100 und die Adressen anderer Server 24).

In Schritt 110 verarbeitet der Agent 30 den URL-Inhalt in den gespeicherten Dateien 32, indem er Server-Verweise durch Verweise der lokalen Datei 32 oder Aufrufe an die lokale Logik ersetzt.

In Schritt 112 werden die Schritte 104 bis 110 für alle Verweise wiederholt und für jeden Durchlauf das nächste URL aktualisiert.

Der Schritt 110 Lokale Logik erstellen macht es erforderlich, dass der Agent 30 eine Wahrheitstabelle 73 oder eine vergleichbare Darstellung aller verschiedener Kombinationen von Antworten A, B, C und D sowie eine Verknüpfung zur Seite, wo sie aufgeführt werden, führt. Dadurch ist es möglich, das Aussehen und die Benutzerführung im Fall mehrerer Eingaben aufrechtzuerhalten, wie aus der Darstellung in Fig. 3 hervorgeht. Der Agent 30 muß lediglich wissen, wo der Server 22 die Verknüpfung, nicht die eigentliche Logik, auflöst, vorausgesetzt, die Verknüpfung ist deterministisch ausgelegt. (Ist diese Voraussetzung nicht erfüllt, ist das Ergebnis ein Schnappschuß, so dass der primäre Nutzen des Verfahrens der vorliegenden Erfindung nur in deterministischen Fällen gilt, also in solchen Fällen, in denen die Server nicht schnell ihre dynamischen Verknüpfungen ändern.) Ein Beispiel für diese deterministische Verwendung ist in einem Entfernungslernsystem anwendbar, in dem die richtigen Antworten ermittelt werden und sich in Abhängigkeit der Zeit oder einer anderen Variablen nicht ändern.

Ein weiterer Vorteil der vorliegenden Erfindung besteht darin, dass ein System und ein Verfahren bereitgestellt werden, um Funktionskomponenten in dynamisch verknüpften interaktiven Web-Seiten so zu ändern, dass sie sich auf der Client-Seite gleich verhalten, ohne dass auf der Server-Seite Transaktionen verarbeitet werden müssen.

Ein weiterer Vorteil der vorliegenden Erfindung besteht darin, dass ein System und ein Verfahren zur Veröffentlichung hoch interaktiver Web-Inhalte auf einem verteilbaren Medium bereitgestellt werden, wodurch eine Server- oder Netzwerkverbindung überflüssig wird.

Ein weiterer Vorteil der vorliegenden Erfindung besteht darin, dass ein System und ein Verfahren für den interaktiven Zugriff auf den Web-Inhalt im Offline-Modus oder dort, wo die Netzwerk-Infrastruktur eine Verbreitung auf lokalen Medien erfordert, bereitgestellt werden.

Ein weiterer Vorteil der vorliegenden Erfindung besteht darin, dass ein System und ein Verfahren zur Veröffentlichung der Inhalte von HTML-Seiten bereitgestellt werden, die auf der Grundlage einer interaktiven Aktion durch den Benutzer von einem Web-Server dynamisch erstellt wurden, als ob diese Daten interaktiv geladen wurden. Ein weiterer Vorteil der vorliegenden Erfindung besteht darin, dass die daraus resultierenden Inhalte über lokale Medien verfügbar gemacht werden können.

Ein weiterer Vorteil der vorliegenden Erfindung besteht darin, dass ein System und ein Verfahren für den Zugriff auf die Inhalte von HTML-Seiten bereitgestellt werden, die auf der Grundlage einer interaktiven Aktion durch den Benutzer von einem Web-Server dynamisch erstellt wurden, ohne dass eine Verbindung zum Server aufgebaut ist.

Ein weiterer Vorteil der vorliegenden Erfindung besteht darin, dass ein System und ein Verfahren zur Veröffentlichung der Inhalte von Web-Seiten auf einer CD-ROM oder auf einem anderen Speichermedium beim Client sowie für den Zugriff auf diese Inhalte über einen nicht angeschlossenen Computer-Browser bereitgestellt werden können.

Ein weiterer Vorteil der vorliegenden Erfindung besteht darin, dass ein System und ein Verfahren bereitgestellt werden, die dynamische Verknüpfungen ermöglichen, die auf der Server-Seite auf der Programmlogik Java oder Common Gateway Architecture (CGA) beruhen.

Ein weiterer Vorteil der vorliegenden Erfindung besteht darin, dass ein System und ein Verfahren bereitgestellt werden, die in der Lage sind, datengesteuerte dynamische Verknüpfungen herzustellen und die ursprüngliche Seite so abzuändern, dass sie alle für den Zugriff auf die verknüpften Seiten erforderlichen Daten enthält.

Ein weiterer Vorteil der vorliegenden Erfindung besteht darin, dass ein System und ein Verfahren bereitgestellt wer-

den, die es einem Client-Web-Cache-Programm ermöglichen, datengesteuerte dynamische Verknüpfungen herzustellen, also Verknüpfungen, die abgeleitet werden, indem auf dem Server eine Logik ausgeführt wird, möglicherweise in Verbindung mit Parametern, die in einer Interaktion zwischen Benutzer und einer Web-Seite übertragen werden, anschließend die ursprüngliche Seite so abzuändern, dass alle benötigten Daten enthalten sind, darunter auch statische Verknüpfungen und Javascript, um zu Seiten zu gelangen, zu denen Verknüpfungen hergestellt wurden.

Ein weiterer Vorteil der vorliegenden Erfindung besteht darin, dass ein System und ein Verfahren für die Umwandlung einer HTML-Gruppe (Hyper-Text Markup Language) bereitgestellt werden, für die eine interaktive Server-Verbindung zu einer HTML-Gruppe erforderlich ist, die keine Server-Interaktion erfordert.

Alternative Ausführungsbeispiele

Zwar wurden in diesem Dokument zur Veranschaulichung ganz spezifische Ausführungsbeispiele der vorliegenden Erfindung beschrieben, doch sei darauf hingewiesen, dass verschiedene Änderungen möglich sind, ohne vom Grundprinzip der vorliegenden Erfindung abzuweichen. Insbesondere liegt es im Anwendungsbereich der vorliegenden Erfindung, ein Computerprogramm oder ein Programmelement oder einen Programmspeicher oder eine Speichereinrichtung wie beispielsweise ein festes oder flüssiges Übertragungsmedium, ein magnetisches oder optisches Kabel, ein Band oder eine Disc oder ähnliches zur Speicherung von maschinenlesbaren Signalen, zur Steuerung der Arbeitsabläufe eines Computers entsprechend dem Verfahren der vorliegenden Erfindung einzubeziehen und/oder ihre Komponenten in Übereinstimmung mit dem System der vorliegenden Erfindung zu strukturieren.

Darüber hinaus läßt sich jeder Verfahrensschritt der vorliegenden Erfindung auf jedem allgemeinen Computer durchführen, beispielsweise auf einem IBM System 390, AS/400, PC oder ähnlichem. Die Verfahrensschritte basieren auf einem oder mehreren Programmelementen, Modulen oder Objekten oder Teilen davon, die mit Hilfe einer Programmiersprache wie C++, Java, P1/1, Fortran o. ä. entwickelt wurden. Ferner kann jeder der genannten Schritte oder eine Datei oder ein Objekt o. ä. mit einer Implementierung des genannten Schritts von einer Spezialhardware oder einem zu diesem Zweck entwickelten Schaltmodul ausgeführt werden.

Demgemäß ist der Anwendungsbereich der vorliegenden Erfindung lediglich durch die nachfolgenden Ansprüche und ihre Entsprechungen begrenzt.

Patentansprüche

1. Verfahren für die Bereitstellung des Zugriffs auf den Inhalt mehrerer Web-Seiten auf einem Server, ohne dass eine Verbindung zum genannten Server besteht, wobei die genannten Web-Seiten dynamische Links aufweisen und das Verfahren folgende Schritte umfaßt:

Anhand des genannten Servers Bestimmung einer dynamischen Verknüpfungsadresse für jede mögliche Kombination aus dynamischen Verknüpfungen auf einer Web-Seite;

Änderung der Web-Seite mit den genannten dynamischen Verknüpfungen, so dass auch entsprechende statische Verknüpfungsadressen aufgenommen werden; und danach

Bildung einer Gruppe von Web-Seiten, die die genannten statischen Verknüpfungsadressen auf einem vom Client zugänglichen Datenspeichermedium entsprechend der genannten Mehrzahl an nicht veränderten Web-Seiten enthält.

2. Verfahren zur Auflösung und Speicherung dynamischer Verknüpfungen als statische Verknüpfungen, wobei dieses Verfahren die folgenden Schritte umfaßt:

Anforderung und Speicherung einer vom Server erzeugten Web-Seite;

Analyse der genannten vom Server erzeugten Web-Seite, um die genannten dynamischen Verknüpfungen zu ermitteln; und

Austausch der genannten dynamischen Verknüpfungen gegen statische Verknüpfungen in einer lokalen Datei, die der genannten Web-Seite entspricht.

3. Verfahren gemäß Anspruch 2, das weiterhin die folgenden Schritte umfaßt:

Speicherung der genannten lokalen Datei im permanenten Speicher; und

Einsatz eines Client-Browsers, um auf die genannte lokale Datei zuzugreifen.

4. Verfahren gemäß Anspruch 2, das weiterhin die folgenden Schritte umfaßt:

Wiederholung der genannten Schritte Anforderung, Speicherung und Analyse und Austausch von Schritten für jede der Server erzeugten Web-Seiten; und

Speicherung der genannten Mehrzahl an Web-Seiten in einem Speicher als Gruppe statisch verknüpfter Web-Seiten, die für einen Client-Browser ohne weiteren Verweis auf den genannten Server verfügbar sind.

5. Verfahren gemäß Anspruch 4, wobei es sich bei dem genannten Speicher um einen permanenten Speicher handelt.

6. Verfahren zur Veröffentlichung dynamisch verknüpfter interaktiver Web-Inhalte als statisch verknüpfte Web-Hierarchie an einer Client-Station, das die folgenden Schritte umfaßt:

Anforderung und Speicherung einer durch den Server erzeugten Web-Seite an einer Client-Station;

Analyse der genannten durch den Server erzeugten Web-Seite, um alle dynamischen Verknüpfungen zu ermitteln;

Wiederholung der genannten Schritte zur Anforderung, Speicherung und Analyse für jede dynamisch verknüpfte Web-Seite, die vom genannten Server bedient wird; und

Austausch der genannten dynamischen Verknüpfungen gegen statische Verknüpfungen in mehreren hierarchisch verknüpften lokalen Dateien am genannten Client entsprechend den genannten durch den Server erzeugten Web-Seiten.

7. Verfahren zur Veröffentlichung dynamisch verknüpfter interaktiver Inhalte einer Web-Seite, das die folgenden Schritte umfaßt:

Erkennung der Struktur der genannten Web-Seite; und

Umwandlung der dynamisch erzeugten Inhalte der genannten Web-Seite in statische Seiten.

8. Verfahren gemäß Anspruch 7, wobei die genannten Schritte zur Erkennung und Umwandlung weiterhin folgendes umfassen: Änderung dynamisch verknüpfter interaktiver Inhalte einer Web-Seite, so dass an einer Client-Station ohne weitere Transaktionsverarbeitung durch den Server ein gleiches Verhalten gewährleistet ist. 5

9. Verfahren gemäß Anspruch 8, das weiterhin die folgenden Schritte umfaßt: Veröffentlichung der resultierenden geänderten dynamisch verknüpften interaktiven Inhalte einer Web-Seite auf einem verbreitbaren Medium.

10. Verfahren gemäß Anspruch 9, das weiterhin die folgenden Schritte umfaßt: Lokale interaktive Kommunikation mit den genannten dynamisch verknüpften interaktiven Inhalten einer Web-Seite, ohne dass eine online-Verbindung zum genannten Server besteht. 10

11. Verfahren für den Zugriff auf den Inhalt von Seiten, die von einem Web-Server auf der Grundlage einer Interaktion des Benutzers ohne online-Verbindung zum genannten Server dynamisch erzeugt wurden, wobei dieses Verfahren weiterhin die folgenden Schritte umfaßt:

Einholung der Adresse des genannten Servers;

Einstellung eines nächsten Suchwerts, der mit der Adresse des genannten Servers korrespondiert; 15

Öffnung einer Verbindung zum genannten nächsten Suchwert;

Anforderung und Speicherung des Inhalts in einen Speicher, auf den der genannte nächste Suchwert zugegriffen hat;

Analyse des genannten Speichers, um alle Verweise auf Suchwerte zu ermitteln;

Wiederholung der genannten Schritte Öffnen, Anforderung, Speicherung und Analyse für jeden der genannten Suchwerte, während der genannte Inhalt, auf den von jedem der genannten Suchwerte zugegriffen wird, in eine entsprechende lokale Datei gespeichert wird; und 20

Verarbeitung der in den genannten lokalen Dateien gespeicherten Inhalte, um die genannten Suchwerte durch lokale Dateiverweise zu ersetzen.

12. Verfahren für den Zugriff auf den Inhalt von Seiten, die von einem Web-Server auf der Grundlage einer Interaktion eines Benutzers ohne online-Verbindung zum genannten Server dynamisch erzeugt wurden, wobei dieses Verfahren weiterhin die folgenden Schritte umfaßt: 25

Einholung der Adresse des genannten Servers;

Einstellung eines nächsten Suchwerts, der mit der Adresse des genannten Servers korrespondiert;

Öffnung einer Verbindung zum genannten nächsten Suchwert; 30

Anforderung und Speicherung des Inhalts in einen Speicher, auf den der genannte nächste Suchwert zugegriffen hat;

Analyse des genannten Speichers, um alle Verweise auf Suchwerte zu ermitteln;

Verarbeitung der im Speicher gespeicherten Inhalte, um die genannten Suchwerte durch lokale Dateiverweise zu ersetzen; 35

Wiederholung der genannten Schritte Öffnen, Anforderung, Speicherung, Analyse und Verarbeitung für jeden der genannten Suchwerte.

13. Verfahren gemäß Anspruch 11, das weiterhin den Schritt der Veröffentlichung der genannten lokalen Dateien auf einem Speichermedium des Clients umfaßt, wobei die genannten Suchwerte durch lokale Dateiverweise ersetzt werden, um auf einen Browser zugreifen zu können, der nicht mit dem genannten Server verbunden ist. 40

14. Verfahren gemäß Anspruch 12, das weiterhin den Schritt der Veröffentlichung der genannten Inhalte auf einem Speichermedium des Clients umfaßt, um den Zugriff eines Computer-Browsers zu ermöglichen, der nicht mit dem genannten Server verbunden ist.

15. Verfahren zur Verarbeitung von Server erzeugten datengesteuerten dynamischen Verknüpfungen, das folgende Schritte umfaßt: 45

Ausführung einer Server-Logik an den Server-gestützten dynamischen Verknüpfungen, die die vom Benutzer eingegebenen und von einem Agent am Client zur Verfügung bereitgestellten Parameter verarbeitet, um den Inhalt einer Web-Seite zu erzeugen;

Speicherung des genannten Inhalts der Web-Seite in lokale Dateien; und

Austausch dynamischer Verknüpfungen im genannten Inhalt der Web-Seite gegen lokale Dateiverweise. 50

16. Verfahren gemäß Anspruch 15, wobei die genannten lokalen Dateiverweise statische Verknüpfungen und eine Logik zur Verarbeitung des Inhalts einer Web-Seite, der in den genannten lokalen Dateien gespeichert ist, enthalten.

17. Verfahren zur Umwandlung einer Server-HTML-Sprache, für die eine Server-Interaktion mit einer Client-HTML-Sprache erforderlich ist, für die keine Server-Interaktion erforderlich ist, wobei dieses Verfahren die folgenden Schritte umfaßt: 55

Ausführung einer Server-Logik auf dem genannten Server, die Benutzerparameter verarbeitet, die von einem Clientgestützten Agent bereitgestellt werden, um die genannte Server-HTML-Sprache zu erzeugen;

Speicherung der genannten Server-HTML-Sprache im genannten Client; und

Austausch dynamischer Verknüpfungen im genannten Server gegen lokale Dateiverweise, um die genannte Client-HTML-Sprache zu erzeugen. 60

18. Ein Agent zur Umwandlung einer Server-HTML-Sprache, für die eine Server-Interaktion zu einer Client-HTML-Sprache erforderlich ist, die keine Server-Interaktion erfordert, umfassend;

Der Agent ist in der Lage, dem genannten Server Benutzerparameter zur Ausführung von Server-Logik auf dem genannten Server bereitzustellen, um die genannte Server-HTML-Sprache zu erzeugen;

Einen Speicher auf Seiten des Clients zum Speichern der genannten Server-HTML-Sprache; und 65

Der Agent ist weiterhin in der Lage, dynamische Verknüpfungen im genannten Server durch lokale Dateiverweise zu ersetzen, um die genannte Client-HTML-Sprache zu erzeugen.

19. System zur Auflösung und Speicherung dynamischer Verknüpfungen als statische Verknüpfungen, das einen

Client-Agent umfaßt, um eine vom Server erzeugte Web-Seite anzufordern und zu speichern, und das weiterhin die Analyse der genannten vom Server erzeugten Web-Seite umfaßt, um die genannten dynamischen Verknüpfungen zu suchen, und das weiterhin den Austausch der genannten dynamischen Verknüpfungen durch statische Verknüpfungen in einer lokalen Datei, die der genannten Web-Seite entspricht, umfaßt.

5 20. System gemäß Anspruch 19, wobei der genannte Client-Agent weiterhin die genannte lokale Datei in einen permanenten Speicher speichert, wobei das System weiterhin einen Client-Browser für den Zugriff auf die genannte lokale Datei umfaßt.

21. System gemäß Anspruch 19, wobei der genannte Client auf jeder vom Server erzeugten Web-Seite dynamische Verknüpfungen iterativ anfordern, speichern, analysieren und durch statische Verknüpfungen ersetzen kann und weiterhin einen lokalen Speicher zur Speicherung der genannten Mehrzahl an Web-Seiten als Gruppe statisch verknüpfter Web-Seiten umfaßt, die einem Client-Browser ohne weiteren Verweis auf den genannten Server zur Verfügung stehen.

22. System gemäß Anspruch 21, wobei der genannte lokale Speicher ein permanenter Speicher ist.

15 23. System für den Zugriff auf den Inhalt von Seiten, die von einem Web-Server auf der Grundlage einer interaktiven Kommunikation des Benutzers ohne online-Verbindung zum genannten Server dynamisch erzeugt wurden, wobei dieses System weiterhin die folgenden Schritte umfaßt:

Mittel zur Einholung der genannten Adresse vom genannten Server;

Mittel zur Einstellung eines nächsten Suchwerts, der mit der Adresse des genannten Servers übereinstimmt;

Mittel zur Öffnung einer Verbindung zum genannten nächsten Suchwert;

20 Mittel zur Anforderung und Speicherung des Inhalts, auf den der genannte nächste Suchwert zugegriffen hat;

Mittel zur Analyse des genannten Speichers, um alle Verweise auf Suchwerte zu ermitteln;

Mittel für iteratives Öffnen, Anfordern, Speichern und Analysieren jedes genannten Suchwerts, während der genannte Inhalt, auf den von jedem der genannten Suchwerte zugegriffen wird, in eine entsprechende lokale Datei gespeichert wird; und

25 Mittel zur Verarbeitung der in den genannten lokalen Dateien gespeicherten Inhalte, um die genannten Suchwerte durch lokale Dateiverweise zu ersetzen.

24. System für den Zugriff auf den Inhalt von Seiten, die von einem Web-Server auf der Grundlage einer interaktiven Kommunikation des Benutzers ohne online-Verbindung zum genannten Server dynamisch erzeugt wurden, wobei dieses System weiterhin die folgenden Schritte umfaßt:

30 Mittel zur Einholung der genannten Adresse vom genannten Server;

Mittel zur Einstellung eines nächsten Suchwerts, der mit der Adresse des genannten Servers übereinstimmt;

Mittel zur Öffnung einer Verbindung zum genannten nächsten Suchwert;

Mittel zur Anforderung und Speicherung des Inhalts, auf den der genannte nächste Suchwert zugegriffen hat;

Mittel zur Analyse des genannten Speichers, um alle Verweise auf Suchwerte zu ermitteln;

35 Mittel zur Verarbeitung der im Speicher gespeicherten Inhalte, um die genannten Suchwerte durch lokale Dateiverweise zu ersetzen; und

Mittel zur Wiederholung der Schritte Öffnen, Anforderung, Speicherung, Analyse und Verarbeitung für jeden der genannten Suchwerte.

40 25. Maschinenlesbare Programmspeichereinrichtung, die ein Befehlsprogramm enthält, das von einem Rechner ausgeführt werden kann, wobei auf einem Server Verfahrensschritte zur Bereitstellung des Zugriffs auf den Inhalt mehrerer Web-Seiten durch einen Client durchgeführt werden, ohne dass eine online-Verbindung zum genannten Server besteht, wobei die genannten Web-Seiten dynamische Verknüpfungen aufweisen und die genannten Verfahrensschritte folgendes umfassen:

Anforderung und Speicherung einer auf dem Server erzeugten Web-Seite an der genannten Client-Station;

45 Analyse der genannten auf dem Server erzeugten Web-Seite, um alle der genannten dynamischen Verknüpfungen zu ermitteln;

Wiederholung der genannten Schritte zur Anforderung, Speicherung und Analyse für jede dynamisch verknüpfte Web-Seite, die vom genannten Server bedient wird; und

50 Austausch der genannten dynamischen Verknüpfungen gegen statische Verknüpfungen in mehreren hierarchisch verknüpften lokalen Dateien am genannten Client entsprechend den genannten auf dem Server erzeugten Web-Seiten.

26. Produktionsartikel, der folgendes umfaßt:

55 ein in einem Computer verwendbares Medium, in dem ein computerlesbares Programmcodemittel integriert ist, um den Zugriff auf den Inhalt mehrerer Web-Seiten eines Servers bereitzustellen, ohne dass zum genannten Server eine online-Verbindung bestehen muß, wobei die Web-Seiten dynamische Verknüpfungen aufweisen und das computerlesbare Programmmittel im genannten Produktionsartikel folgendes umfaßt:

ein computerlesbares Programmcodemittel, das einen Computer auffordert, eine vom Server erzeugte Web-Seite am genannten Client anzufordern und zu speichern;

60 ein computerlesbares Programmcodemittel, das einen Computer auffordert, die genannte vom Server erzeugte Web-Seite auf alle genannten dynamischen Verknüpfungen zu durchsuchen;

ein computerlesbares Programmcodemittel, das einen Computer auffordert, die genannten Schritte des Anforderns, Speicherns und Analysierens für jede dynamisch verknüpfte Web-Seite, die vom genannten Server bedient wird, zu wiederholen; und

65 ein computerlesbares Programmcodemittel, das einen Computer auffordert, die genannten dynamischen Verknüpfungen durch statische Verknüpfungen in mehreren hierarchisch verknüpften lokalen Dateien am genannten Client entsprechend den vom genannten Server erzeugten Web-Seiten zu ersetzen.

27. Computerprogrammprodukt, das so konfiguriert ist, dass es dynamische HTML-Verknüpfungen (Hypertext Markup Language) durch errechnete statische Darstellungen gemäß folgenden Schritten ersetzt:

Ausführung einer Server-Logik mit einer Server-Gruppe von HTML-Verknüpfungen abhängig von Benutzerparametern, die von einem Agent bereitgestellt werden, um die genannte Server-Gruppe der HTML zu erzeugen; Speicherung der genannten Server-Gruppe der HTML; und Ersetzen der dynamischen Verknüpfungen in der genannten Server-Gruppe durch lokale Dateiverweise, um die genannte Client-Gruppe der HTML zu erzeugen.

28. Maschinenlesbare Programmspeichereinrichtung, die ein Befehlsprogramm umfaßt, das von einem Rechner ausgeführt werden kann, um die genannten Verfahrensschritte für den Zugriff auf den Inhalt von Seiten, die von einem Web-Server auf der Grundlage einer interaktiven Kommunikation mit dem Benutzer dynamisch erzeugt wurden, durchzuführen, ohne dass eine Verbindung zum genannten Server besteht, wobei die genannten Verfahrensschritte folgende sind:

Einholung der Adresse des genannten Servers;

Einstellung eines nächsten Suchwerts, der mit der Adresse des genannten Servers übereinstimmt;

Öffnung einer Verbindung zum genannten nächsten Suchwert;

Anforderung und Speicherung des Inhalts, auf den der genannte nächste Suchwert zugegriffen hat;

Analyse des genannten Speichers, um alle Verweise auf Suchwerte zu ermitteln;

Wiederholung der genannten Schritte Öffnen, Anforderung, Speicherung und Analyse für jeden der genannten Suchwerte, während der genannte Inhalt, auf den von jedem der genannten Suchwerte zugegriffen wird, in eine entsprechende lokale Datei gespeichert wird; und

Verarbeitung der in den genannten lokalen Dateien gespeicherten Inhalte, um die genannten Suchwerte durch lokale Dateiverweise zu ersetzen.

29. Maschinenlesbare Programmspeichereinrichtung, die ein Befehlsprogramm enthält, das von einem Rechner ausgeführt werden kann, wobei Verfahrensschritte durchgeführt werden, um auf den Inhalt von Web-Seiten zuzugreifen, die von einem Web-Server auf der Grundlage einer interaktiven Kommunikation mit dem Benutzer dynamisch erzeugt wurden, ohne dass Verbindung zum genannten Server besteht, wobei die genannten Verfahrensschritte folgendes umfassen:

Einholung der Adresse des genannten Servers;

Einstellung eines nächsten Suchwerts, der mit der Adresse des genannten Servers übereinstimmt;

Öffnung einer Verbindung zum genannten nächsten Suchwert;

Anforderung und Speicherung des Inhalts, auf den der genannte nächste Suchwert zugegriffen hat;

Analyse des genannten Speichers, um alle Verweise auf Suchwerte zu ermitteln;

Verarbeitung der gespeicherten Inhalte, um die genannten Suchwerte durch lokale Dateiverweise zu ersetzen; und Wiederholung der genannten Schritte Öffnen, Anforderung, Speicherung, Analyse und Verarbeitung für jeden der genannten Suchwerte.

30. Programmspeichereinrichtung gemäß Anspruch 29, wobei die genannten Verfahrensschritte weiterhin den Schritt der Weiterleitung der genannten lokalen Dateien an ein Speichermedium am Client umfassen, wobei die genannten Suchwerte durch lokale Dateiverweise für den Zugriff eines Browsers ersetzt werden, der nicht mit dem genannten Server verbunden ist.

31. Verfahren für den Zugriff eines Clients auf die Inhalte von Seiten, die auf der Grundlage einer interaktiven Aktion durch den Benutzer von einem Web-Server dynamisch erstellt wurden, ohne dass eine Verbindung zwischen dem genannten Client und dem genannten Server aufgebaut ist, wobei dieses Verfahren die folgenden Schritte umfaßt:

der genannte Client ist mit dem genannten Server verbunden,

Simulation der interaktiven Benutzeraktion und Durchforstung dynamischer Web-Seiten, um Server-Prozesse einzuleiten, die Web-Seiten an den genannten Client weiterleiten; und

Ansammlung der genannten Web-Seiten, während sie bereitgestellt werden;

Änderung der angesammelten Web-Seiten, indem statische Hyperlinks aufgenommen werden, die auf Server-Seite eine gerichtete Navigationslogik ersetzen; und

Permanente Speicherung der genannten angesammelten Web-Seiten mit den genannten statischen Hyperlinks in lokalen Dateien, wo sie für die Offline-Navigation durch einen Client-Browser verfügbar sind.

Hierzu 6 Seite(n) Zeichnungen

- Leerseite -

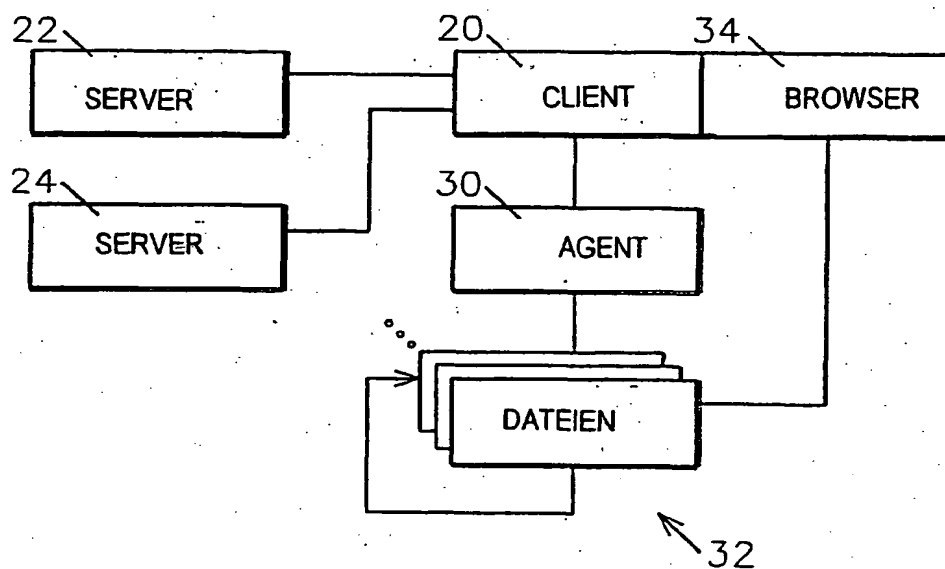


FIG. 1

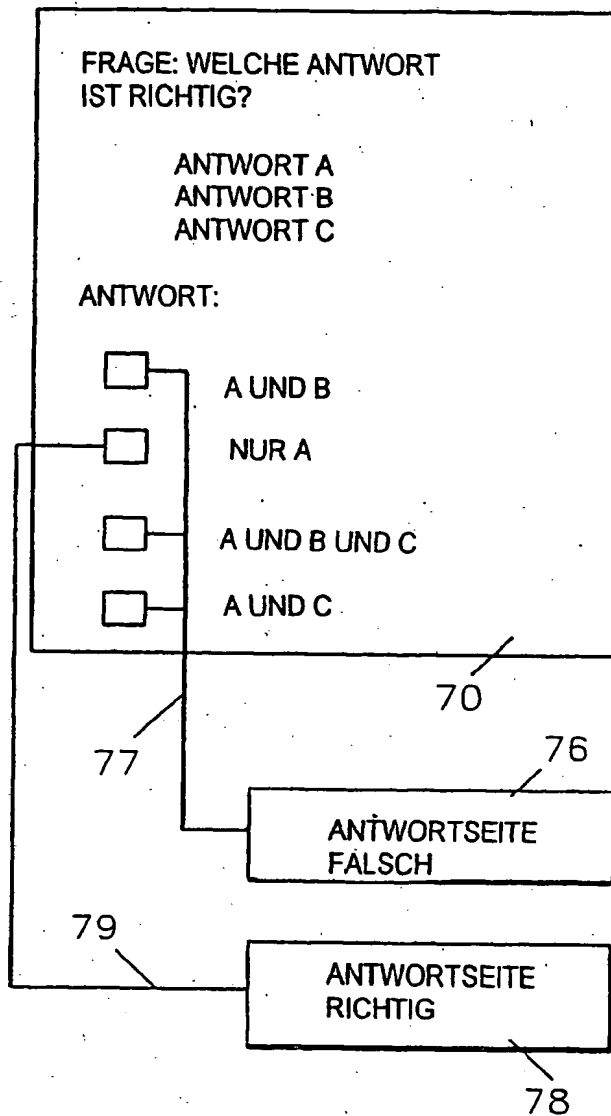


FIG. 2

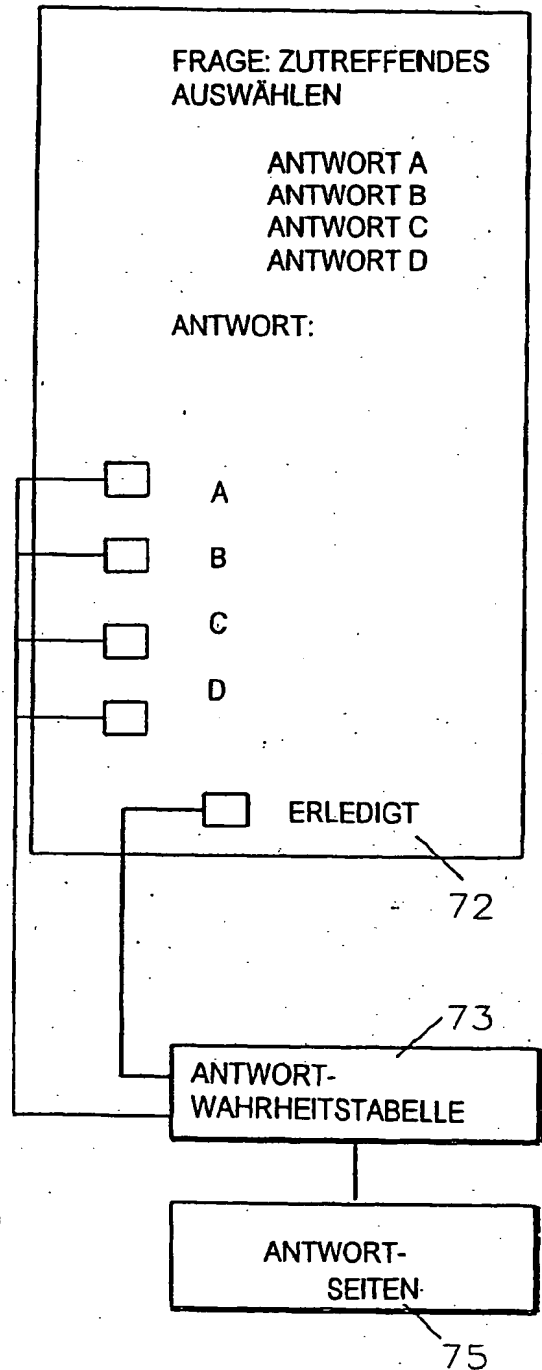


FIG. 3

FRAGE: WÄHLEN SIE
EINE ANTWORT AUS:

ANTWORT A
ANTWORT B
ANTWORT C
ANTWORT D

ANTWORT:

<input type="checkbox"/>	KEINE
<input type="checkbox"/>	NUR A
<input type="checkbox"/>	NUR B
<input type="checkbox"/>	NUR C
<input type="checkbox"/>	NUR D
<input type="checkbox"/>	A UND B
<input type="checkbox"/>	A UND C
<input type="checkbox"/>	A UND D
<input type="checkbox"/>	B UND C
<input type="checkbox"/>	B UND D
<input type="checkbox"/>	C UND D
<input type="checkbox"/>	A UND B UND C
<input type="checkbox"/>	A UND B UND D
<input type="checkbox"/>	A UND C UND D
<input type="checkbox"/>	B UND C UND D
<input type="checkbox"/>	A UND B UND C UND D
<input type="checkbox"/>	

74

FIG. 4

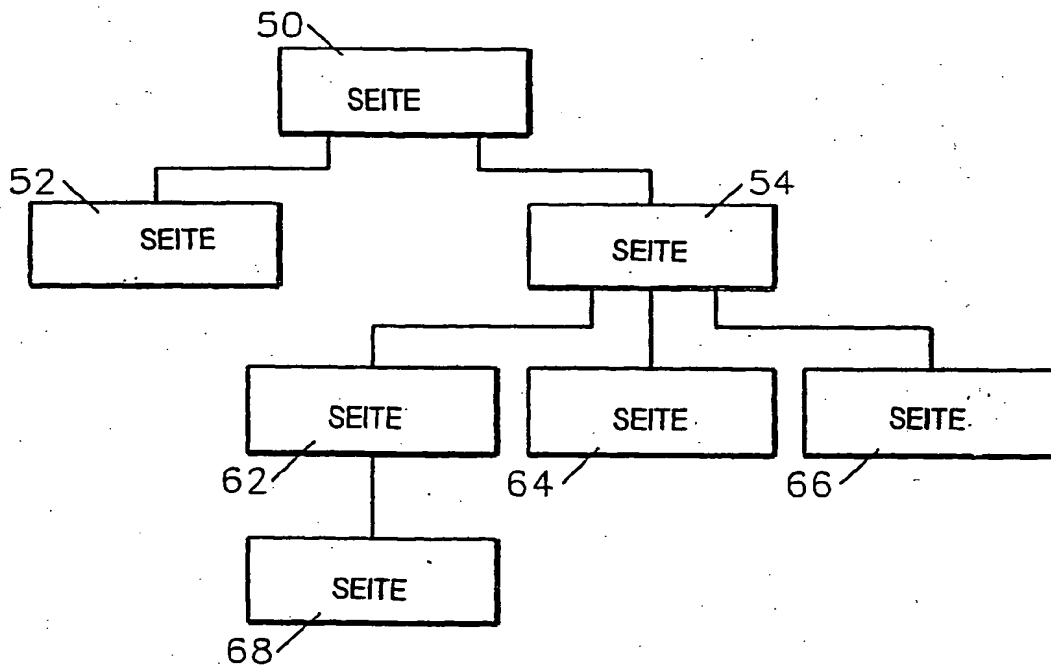


FIG. 5

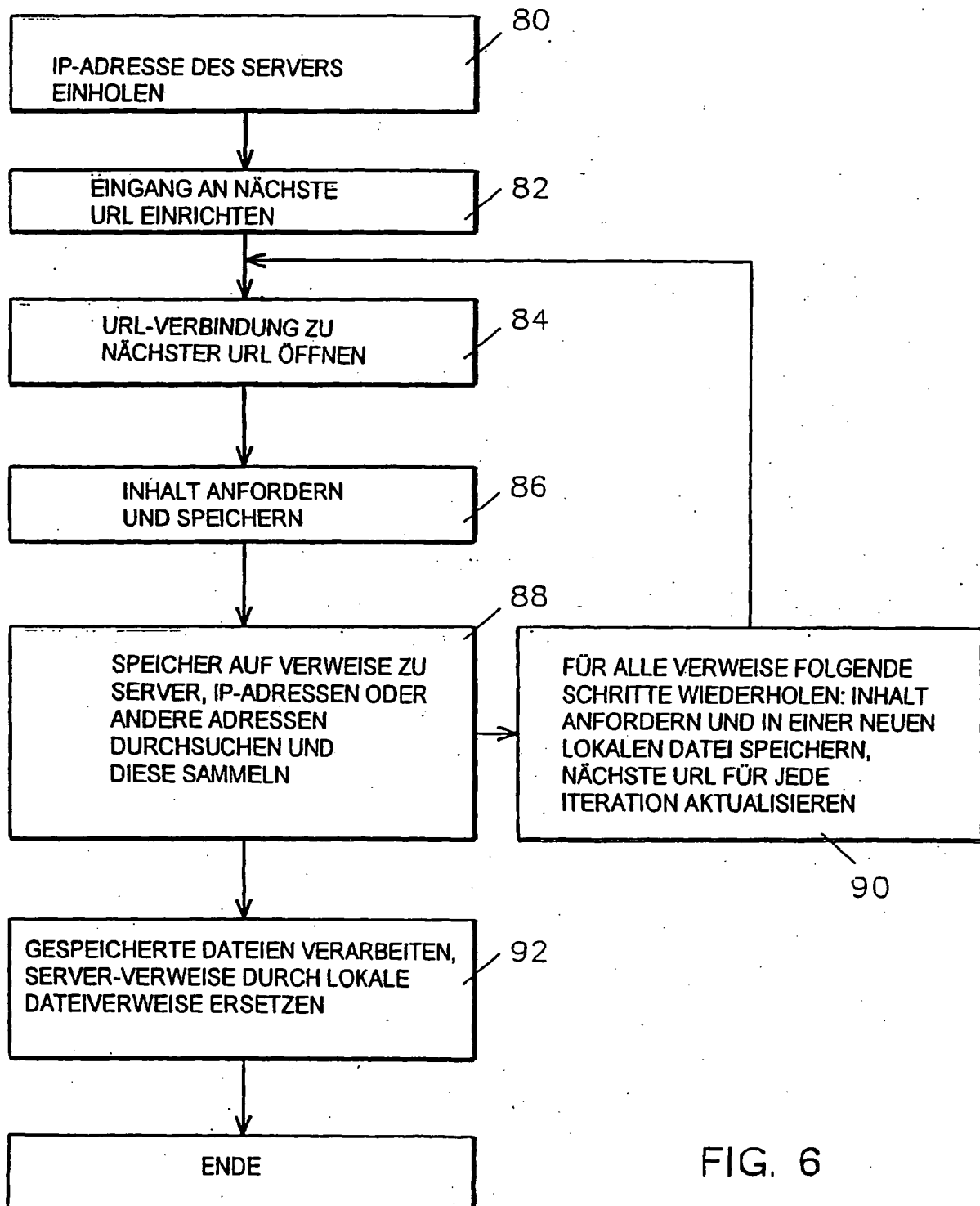


FIG. 6

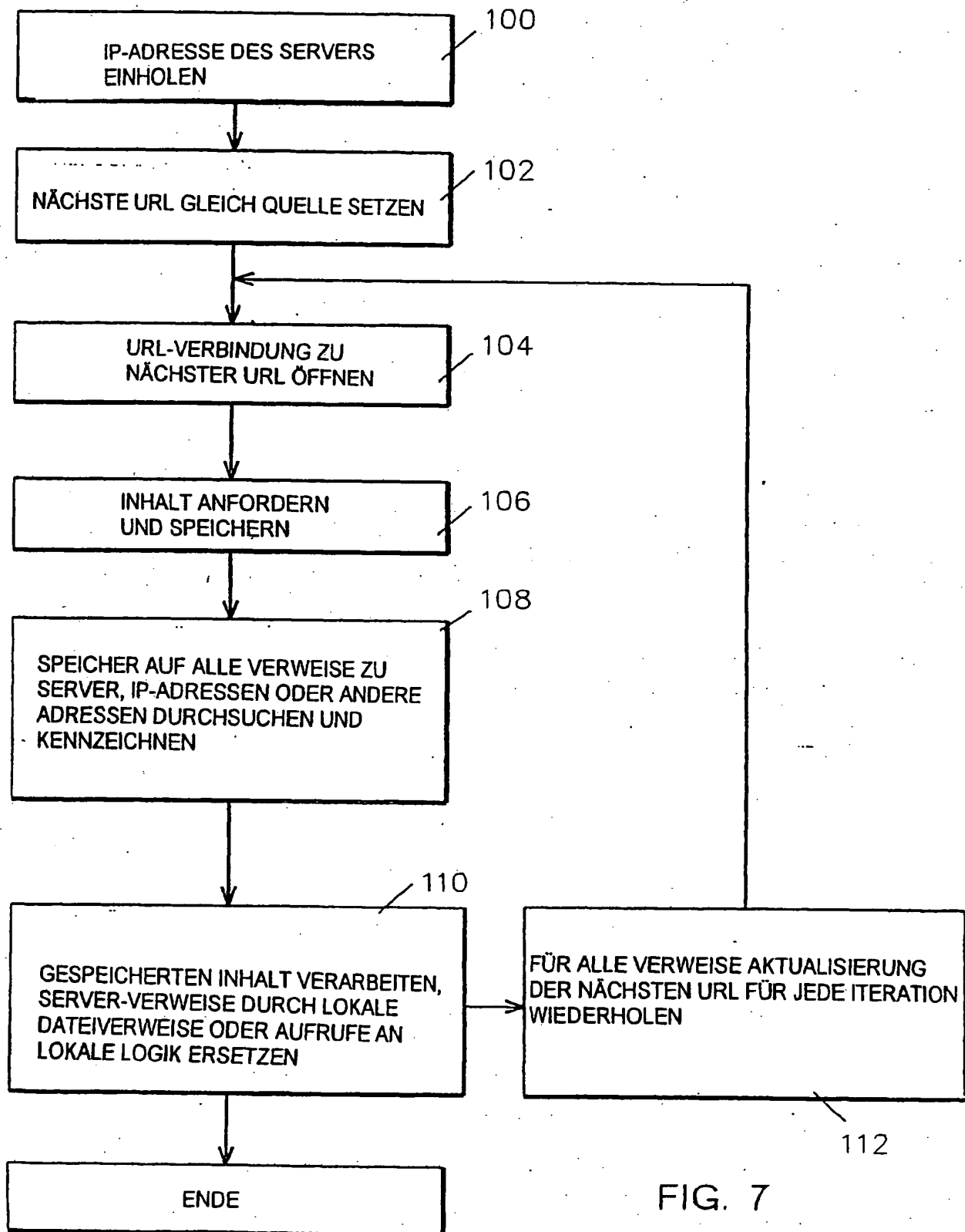


FIG. 7